Semismooth Support Vector Machines*

Michael C. Ferris†
Todd S. Munson‡

November 29, 2000

Abstract

The linear support vector machine can be posed as a quadratic program in a variety of ways. In this paper, we look at a formulation using the two-norm for the misclassification error that leads to a positive definite quadratic program with a single equality constraint when the Wolfe dual is taken. The quadratic term is a small rank update to a positive definite matrix. We reformulate the optimality conditions as a semismooth system of equations using the Fischer-Burmeister function and apply a damped Newton method to solve the resulting problem. The algorithm is shown to converge from any starting point with a Q-quadratic rate of convergence. At each iteration, we use the Sherman-Morrison-Woodbury update formula to solve a single linear system of equations. Significant computational savings are realized as the inactive variables are identified and exploited during the solution process. Results for a 60 million variable problem are presented, demonstrating the effectiveness of the proposed method on a personal computer.

1 Introduction

The support vector machine is used to construct a (linear or nonlinear) surface that "optimally" partitions measurements taken from representative subsets of known populations. The surface is then used to "determine" the origins of unknown observations. The technique is one example of a supervised learning process from the machine learning community. We will be considering the case where we have two populations and want to construct a linear surface. One example application is in the determination of malignant and benign tumors [6,

*This work partially supported by NSF grant number CCR-9972372; AFOSR grant number F49620-01-1-0040; the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38; and Microsoft Corporation.

†Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, Wisconsin 53706

‡Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, Illinois 60439

18, 19], which are the two populations. Several models exist for the calculation of an optimal partitioning surface. In this paper, we will consider one such model that leads to a positive definite quadratic program with bounds and a single equality constraint.

Our main goal in this paper is to present an algorithm for solving the resulting optimization problem that converges from any starting point and, near a solution, has a quadratic rate of convergence. For this purpose, we present a semismooth method [4] for solving the optimality conditions for the problem and prove the necessary results. In particular, we show that for every x, all elements of the generalized Jacobian are nonsingular, and the sequence produced by the algorithm contains an accumulation point. We then apply standard theory to show that the algorithm actually converges to a solution at a Q-quadratic rate.

Other algorithms have been proposed for solving these types of problems, including an interior-point method in [7] and an active set method in [17]. The proposed semismooth algorithm solves only linear systems of equations and exploits an "active" set implicitly defined by the algorithm in the linear algebra computation. The resulting semismooth code scales well to large sample populations. We target sample populations with 1-60 million observations. We note that 60 million observations corresponds to a random sampling of 100% of the current population of Britain, 20% of the current population of the United States, and 1% of the current world population. Therefore, we believe the 60 million observation problem to be a reasonable test problem. To achieve scalability, we use the Sherman-Morrison-Woodbury update formula in the calculation of the direction and use asynchronous I/O to retrieve the observation data from disk. The resulting code uses a small number of vectors with memory and disk requirements suitable for a personal computer.

The first section of this paper derives the linear support vector machine formulation we use for the subsequent analysis and testing. The problem is posed as a mixed complementarity problem where the mixed components are equality constraints. Specifically, let $\mathcal{L}$ and $\mathcal{E}$ be a partition of the indices $\{1, 2, \ldots, n\}$, implicitly corresponding to lower bounded and free variables, and let $F : \Re^n \to \Re^n$ be a given function. Let $m = \mathbf{card}\,(\mathcal{L})$ and $c = \mathbf{card}\,(\mathcal{E})$. The mixed complementarity problem considered is to find an $z^*_{\mathcal{L}} \in \Re^m$ and $z^*_{\mathcal{E}} \in \Re^c$ such that

$$\begin{array}{c} 0 \le F_{\mathcal{L}}(z_{\mathcal{L}}, z_{\mathcal{E}}) \;\perp\; z_{\mathcal{L}} \ge 0 \\ F_{\mathcal{E}}(z_{\mathcal{L}}, z_{\mathcal{E}}) = 0, \end{array} \tag{1}$$

where $\perp$ is defined componentwise as $0 \le a \;\perp\; b \ge 0$ if and only if $a \ge 0$, $b \ge 0$, and $ab = 0$. This problem is the standard nonlinear complementarity problem when $c = 0$ and a square system of nonlinear equations when $m = 0$. See [8] for definitions of general mixed complementarity problems and applications. In this paper we concentrate on mixed complementarity problems that are a combination of lower bounded variables and free variables. The extension of the algorithms to cases where the variables are lower and upper bounded is straightforward and is given in [21].

The second section details a damped Newton method for semismooth equations [4, 21] for solving such complementarity problems. The proposed method uses the Fischer-Burmeister function [9] to reformulate the complementarity conditions as a system of semismooth equations. The basic theory for these methods is given with appropriate citations to the literature. We further prove that the method converges when applied to our support vector machine formulation. In particular, we demonstrate how to compute the Newton direction at any arbitrary point using applications of the Sherman-Morrison-Woodbury update formula [22].

The final section discusses an implementation of the method using out-of-core computations. We present some results for a large test problem containing 60 million points and compare them with the interior-point method results given in [7].

2 Linear Support Vector Machine

The linear support vector machine attempts to separate two finite point sets with a hyperplane such that the separation margin is maximized. Before delving into the exact problem formulation, we first present the basic notation used throughout this paper. Consider two populations $\mathcal{P}_+$ and $\mathcal{P}_-$ that have been sampled, and let $P_+ \subseteq \mathcal{P}_+$ and $P_- \subseteq \mathcal{P}_-$ denote finite sample sets and $P \equiv P_+ \cup P_-$ the entire set of sampled elements. Let $m = \mathbf{card}\,(P)$ denote the size of the total population for the remainder of this paper. We associate with each $p \in P$ a vector $a(p) \in \Re^f$ that measures f features for the particular element. Furthermore, let $A(P) \in \Re^{m \times f}$ denote the matrix formed by the measured observations for each $p \in P$, and let $A_+ := A(P_+)$ and $A_- := A(P_-)$.

Assuming the two point sets are disjoint, $(co \cup_{p \in P_+} a(p)) \cap (co \cup_{p \in P_-} a(p)) = \emptyset$, we can select $w \in \Re^f$ and $\gamma \in \Re$ such that $A_+ w < \gamma$ and $A_- w > \gamma$. We note that the hyperplane $\{x \in \Re^f \mid x^T w = \gamma\}$ strictly separates the two point sets and that the separation margin [2, 26, 28], the minimum distance from the hyperplane to the convex hulls of the point sets, is $\frac{2}{\|w\|_2^2}$. Therefore, an optimization problem to maximize the separation margin would be

$$\begin{array}{rl} \max_{w,\gamma} & \frac{2}{\|w\|_2^2} \\ \text{subject to} & A_+ w < \gamma \\ & A_- w > \gamma. \end{array}$$

We note that maximizing $\frac{2}{\|w\|_2^2}$ is the same as minimizing $\frac{1}{2}\|w\|_2^2$ and that the strict inequalities can be removed by normalizing the system [15]. Therefore, we obtain the following quadratic optimization problem:

$$\begin{array}{rl} \min_{w,\gamma} & \frac{1}{2}\|w\|_2^2 \\ \text{subject to} & A_+ w - \gamma e \geq 1 \\ & A_- w - \gamma e \leq -1. \end{array} \tag{2}$$

The constraints can be more succinctly written if we define an "indicator" function, $d(p)$, as follows:

$$d(p) := \begin{cases} 1 & \text{if } p \in P_+ \\ -1 & \text{if } p \in P_- \end{cases}$$

with D denoting the diagonal matrix formed from $d(p)$ for all $p \in P$. Then, we can write the single constraint

$$D(Aw - \gamma e) \geq 1.$$

The reason for maximizing the separation margin is to improve the generalization ability [14] of the computed separating surface.

Unfortunately, the underlying assumption above that the two point sets are disjoint is typically not satisfied, and (2) is infeasible. In this case, a surface is constructed that minimizes the error in satisfying the inequalities, termed the misclassification error in the machine learning community [13]. The resulting optimization problem in this case becomes

$$\begin{array}{ll} \min_{w,\gamma,y} & \frac{1}{2}\|y\|_2^2 \\ & D(Aw - \gamma e) + y \geq 1 \\ & y \geq 0, \end{array} \tag{3}$$

where we have used the two norm of the misclassification error. We note that the constraint $y \geq 0$ is unnecessary and will be dropped from the problem. Other norms can be used for the misclassification error, which lead to other problem formulations.

We now combine the two problems, (2) and (3), by introducing a parameter $\nu > 0$ that weights the two competing goals, maximizing the separation margin and minimizing the misclassification error. The resulting optimization problem, termed a support vector machine, is

$$\begin{array}{ll} \min_{w,\gamma,y} & \frac{1}{2}\|w\| + \frac{\nu}{2}\|y\|_2^2 \\ & D(Aw - \gamma e) + y \geq 1, \end{array} \tag{4}$$

which is a convex quadratic program that is feasible with the objective bounded below by zero. Hence, (4) has a solution, (w^*, γ^*, y^*). The support vectors are the points where $D(Aw^* - \gamma^* e) \leq 1$, that is the misclassified points and the points on the bounding hyperplanes generated.

The Wolfe dual [12] of (4) is the strongly convex quadratic program

$$\begin{array}{ll} \min_{x} & \frac{1}{2\nu}x^T x + \frac{1}{2}x^T DAA^T D^T x - e^T x \\ & e^T D^T x = 0 \\ & x \geq 0 \end{array}, \tag{5}$$

which has a unique solution, x^*. We note that the quadratic term consists of a rank-f update to a positive definite matrix, which will become useful in the algorithm development and necessary linear algebra calculations.

The final step in the problem derivation is to write first order necessary and sufficient optimality conditions for (5), which form the mixed linear complementarity problem:

$$\begin{array}{c} 0 \leq \left(\frac{1}{\nu} I + DAA^T D^T\right) x - De\mu - e \;\perp\; x \geq 0 \\ e^T D^T x = 0. \end{array} \tag{6}$$

Theorem 2.1 *Let* $\mathbf{card}\,(P_+) > 0$ *and* $\mathbf{card}\,(P_-) > 0$. *Then (6) has a unique solution.*

Proof: Since (5) is a strongly convex quadratic program that is feasible and bounded below, it has a unique solution. Let x^* denote this solution. Furthermore, there must exist a μ^* such that (x^*, μ^*) is a solution to (6). The remainder of this proof shows that μ^* is unique.

Assume that $x^* = 0$. Therefore, any μ^* solving (6) must satisfy $-De\mu^* - e \geq 0$. Recalling the definition of D and using the fact that $\mathbf{card}\,(P_+) > 0$ and $\mathbf{card}\,(P_-) > 0$ by assumption, we have $\mu^* \leq -1$ and $\mu^* \geq 1$, a contradiction. Therefore, $x^* \neq 0$.

Since x^* solves (5) and $x^* \neq 0$, we must have that $x_i^* > 0$ for some i, since x^* must also be feasible for (5). Therefore, any μ^* solving (6) must satisfy $\left[\left(\frac{1}{\nu} I + DAA^T D^T\right) x^* - De\mu^* - e\right]_i = 0$. Hence, μ^* is uniquely determined by this equation, and the proof is complete.
Q.E.D.

3 Algorithm

The preceding section developed the support vector machine formulation we consider. This section develops a semismooth method [4, 24] based on the Fischer-Burmeister merit function [9] in order to solve the linear mixed complementarity problems defined by the necessary and sufficient first order optimality conditions found in (6). Essentially, the semismooth method reformulates this complementarity problem as a system of nonlinear, nonsmooth equations and applies a generalized Newton method to find a solution. We start by defining a basic semismooth method and the convergence theory and then discuss the Fischer-Burmeister function and its properties. Finally, we specialize the method for the support vector machine problem and prove convergence. The proofs here tell us how to perform the linear algebra in the implementation.

3.1 Basic Semismooth Method

The class of semismooth functions [20, 23, 25] are a generalized notion of continuously differentiable functions that are both Lipschitzian and directionally differentiable. To define the class of semismooth functions precisely, we introduce the notion of the B-subdifferential and generalized Jacobian. Let $G : \Re^n \to \Re^n$

be a Lipschitzian function and D_G denote the set of points where G is differentiable. This definition for D_G is appropriate because, by Rademacher's theorem, G is differentiable almost everywhere.

Before proceeding, we describe some notation used in the sequel. If $F : \Re^n \to \Re^n$, we denote its Jacobian at a point by $F'(x)$ and let $\nabla F(x)$ denote the transposed Jacobian. In particular:

$$\begin{aligned} [F'(x)]_{i,j} &:= \frac{\partial F_i(x)}{\partial x_j} \\ [\nabla F(x)]_{i,j} &:= \frac{\partial F_j(x)}{\partial x_i}. \end{aligned}$$

Furthermore, $F'(x;d)$ will denote the directional derivative of F at x in the direction d. Finally $co\ (X)$ will denote the convex hull of a set X. We then have the following definitions:

Definition 3.1 (B-subdifferential [25]) *The B-subdifferential of G at z is*

$$\partial_B G(z) := \left\{ H \mid \exists \{z^k\} \to z, z^k \in D_G, \text{ and } \lim_{\{z^k\} \to z} G'(z^k) = H \right\}.$$

Definition 3.2 (Generalized Jacobian [3]) *The Clarke generalized Jacobian of G at z is*

$$\partial G(z) := co\ \partial_B G(z).$$

Definition 3.3 (Semismooth) *Let $G : \Re^n \to \Re^n$ be locally Lipschitzian at $z \in \Re^n$. Then G is semismooth at z if:*

$$\lim_{\substack{H \in \partial G(z+td') \\ d' \to d, t \downarrow 0}} Hd' \tag{7}$$

exists for all $d \in \Re^n$. In particular, G is directionally differentiable at z with $G'(z;d)$ given by the limit in (7). If, in addition, for any $d \to 0$ and any $H \in \partial G(z+d)$,

$$Hd - G'(z;d) = O\left(\|d\|^2\right),$$

then G is said to be strongly semismooth at z. Furthermore, G is a (strongly) semismooth function if G is (strongly) semismooth for all $z \in \Re^n$.

We are interested in finding a solution to the system of equations $G(z) = 0$, where $G : \Re^n \to \Re^n$ is a semismooth function. To solve this system we will use a damped Newton method [24]. To this end, we define a merit function $g : \Re^n \to \Re$ as $g(z) := \frac{1}{2}\|G(z)\|_2^2$ and assume that g is continuously differentiable. The algorithm follows.

Algorithm 3.4 (Damped Newton Method for Semismooth Equations)

0. (Initialization) Let $z^0 \in \Re^n$, $\rho > 0$, $p > 2$, and $\sigma \in \left(0, \frac{1}{2}\right)$ be given. Set $k = 0$.

1. (Termination) If $g(z^k) = 0$, stop.

2. (Direction Generation) Otherwise, let $H^k \in \partial_B G(z^k)$, and calculate $d^k \in \Re^n$ solving the Newton system:

$$H^k d^k = -G(z^k). \tag{8}$$

If either (8) is unsolvable or the descent condition

$$\nabla g(z^k)^T d^k < -\rho \left\| d_k \right\|_2^p \tag{9}$$

is not satisfied, then set $d^k = -\nabla g(z^k)$.

3. (Linesearch) Choose $t^k = 2^{-i_k}$, where i_k is the smallest integer such that

$$g\left(z^k + 2^{-i_k} d^k\right) \leq g(z^k) + \sigma 2^{-i_k} \nabla g(z^k)^T d^k. \tag{10}$$

4. (Update) Let $z^{k+1} := z^k + t^k d^k$ and $k := k + 1$. Go to 2.

We then have the following convergence theorem, whose proof can be found in [24, 4]:

Theorem 3.5 *Let $G : \Re^n \to \Re^n$ be semismooth for all $z \in \Re^n$ and $g(z)$ be continuously differentiable. Let $\{z^k\}$ be a sequence generated by Algorithm 3.4. Then any accumulation point of $\{z^k\}$ is a stationary point for $g(z)$. Furthermore, if one of these accumulation points, say z^*, solves the system $G(z) = 0$ and all $H \in \partial_B G(z^*)$ are invertible, then the following hold:*

a. For all k sufficiently large, the Newton direction calculated in (8) exists and satisfies both the descent condition (9) and linesearch rule (10) with $t^k = 1$.

b. $\{z^k\}$ converges to z^* and the rate of convergence is Q-superlinear.

c. If in addition G is strongly semismooth at z^*, then the rate of convergence is Q-quadratic.

3.2 Fischer-Burmeister Function

Complementarity problems such as the one in (6) can be solved by reformulating them as (square) systems of semismooth equations and applying Algorithm 3.4 [4]. We will concentrate on one reformulation using the Fischer-Burmeister function [9]. We define a function $\phi : \Re^2 \to \Re$ as follows:

$$\phi(a, b) := a + b - \sqrt{a^2 + b^2}.$$

This function has the NCP-property that $\phi(a,b) = 0 \Leftrightarrow 0 \leq a \perp b \geq 0$. Therefore, letting $n = m + c$, we can define $\Phi : \Re^n \to \Re^n$ as

$$\Phi(z) := \begin{bmatrix} \phi(z_1, F_1(z)) \\ \vdots \\ \phi(x_m, F_m(z)) \\ F_{m+1}(z) \\ \vdots \\ F_n(z) \end{bmatrix}, \tag{11}$$

where we have m nonlinear complementarity constraints and c equation constraints. We summarize the properties of this function in the following theorem.

Theorem 3.6 ([1]) *Let $F : \Re^n \to \Re^n$ be continuously differentiable. Then the following hold:*

a. *Φ is a semismooth function. If in addition F is twice continuously differentiable with Lipschitz continuous second derivatives, then Φ is a strongly semismooth function.*

b. *$\Psi(z) := \frac{1}{2}\|\Phi(z)\|_2^2$ is continuously differentiable with $\nabla\Psi(z) = H^T\Phi(z)$ for any $H \in \partial_B\Phi(z)$.*

c. *$\Phi(z^*) = 0$ if and only if z^* solves the complementarity problem defined by F.*

d. *If z^* is a stationary point of Ψ and there exists an $H \in \partial_B\Phi(z^*)$ that is invertible, then $\Phi(z^*) = 0$ and hence, z^* solves the complementarity problem defined by F.*

Methods for calculating an element of the B-subdifferential can be found for example in [1, 4]. While we use these constructions in our algorithm, we will be using an overestimate of the B-subdifferential (detailed in the following theorem) for the proofs in the sequel.

Theorem 3.7 ([1, 4]) *Let $F : \Re^n \to \Re^n$ be continuously differentiable. Then*

$$\partial_B\Phi(z) \subseteq \{D_a + D_b F'(z)\},$$

where $D_a \in \Re^{n\times n}$ and $D_b \in \Re^{n\times n}$ are diagonal matrices with entries defined as follows:

a. *For all $i \in \{1, \ldots, m\}$: If $\|(z_i, F_i(z))\| \neq 0$, then*

$$(D_a)_{ii} = 1 - \frac{z_i}{\|(z_i, F_i(z))\|}$$
$$(D_b)_{ii} = 1 - \frac{F_i(z)}{\|(z_i, F_i(z))\|};$$

otherwise

$$((D_a)_{ii}, (D_b)_{ii}) \in \left\{(1-\eta, 1-\rho) \in \Re^2 \;\middle|\; \|(\eta,\rho)\| \leq 1\right\}.$$

b. For all $i \in \{m+1, \ldots, n\}$:

$$(D_a)_{ii} = 0$$
$$(D_b)_{ii} = 1.$$

Furthermore, we have for all $i \in \{1, \ldots, n\}$, $(D_a)_{ii} \geq 0$, $(D_b)_{ii} \geq 0$, and $(D_a)_{ii} + (D_b)_{ii} > 0$. In particular, $D_a + D_b$ is positive definite.

3.3 Support Vector Machine Specialization

We have given an algorithm for solving semismooth systems of equations and have shown how to pose the complementarity problem using the Fischer-Burmeister function as a semismooth system of equations.

At this stage, it would be nice to present standard convergence material that showed Algorithm 3.4 converges to a solution of (6), perhaps with a given rate. Unfortunately, such results are not available. The typical results presented in the literature assume that the equation in (6) can either be explicitly substituted out of the model (which cannot be done in this case) or assume that F is a uniform P-function (which is also not the case here). Instead of using these results, we directly prove the necessary conditions for our particular model.

We now show that Algorithm 3.4 converges when applied to (6). The first step is to establish that for all $z \in \Re^n$, all $H \in \partial_B \Phi(z)$ are invertible. To do this, we will show how to compute the Newton direction from step 2 of Algorithm 3.4.

Recall that we want to solve the complementarity problem (6) using the Fischer-Burmeister reformulation. Therefore, at every iteration of Algorithm 3.4, we solve the system of equations

$$\begin{bmatrix} D_a + D_b(\frac{1}{\nu} I + DAA^T D^T) & -D_b De \\ e^T D^T & 0 \end{bmatrix} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

for some r_1 and r_2 and diagonal matrices D_a and D_b chosen according to Theorem 3.7.

Proposition 3.8 *Suppose the mixed complementarity problem and F are defined as in (1). For all $i \in \{1, \ldots, m\}$, if $(D_b)_{ii} = 0$, then $z_i = 0$ and $F_i(z) \geq 0$.*

Proof: Let i be given with $(D_b)_{ii} = 0$. There are two cases to consider. If $\|(z_i, F_i(z))\| > 0$, then

$$\begin{aligned} 0 = (D_b)_{ii} = 1 - \frac{F_i(z)}{\|(z_i, F_i(z))\|} \quad &\Longrightarrow \quad \frac{F_i(z)}{\|(z_i, F_i(z))\|} = 1 \\ &\Longrightarrow \quad F_i(z) = \|(z_i, F_i(z))\| > 0. \end{aligned}$$

Furthermore, since $F_i(z) > 0$ and $F_i(z) = \|(z_i, F_i(z))\|$, we have that $z_i = 0$. In the other case, $\|(z_i, F_i(z))\| = 0$, which implies $z_i = 0$ and $F_i(z) = 0$. Therefore, the conclusion of the proposition holds in both cases and the proof is complete. **Q.E.D.**

Proposition 3.9 *Let* **card** $(P_+) > 0$ *and* **card** $(P_-) > 0$. *Then for the model considered in (6),* $D_b \neq 0$.

Proof: Assume $D_b = 0$. Then by Proposition 3.8, for all $i \in \{1, \ldots, m\}$, $z_i = 0$ and $F_i(z) \geq 0$. Looking at the definition of $F(0, \mu)$, we then have that $-eD^T\mu - e \geq 0$ with D defined in Section 2. Since **card** $(P_+) > 0$ and **card** $(P_-) > 0$ by assumption, this reduces to a system of two inequalities, $\mu - 1 \geq 0$ and $-\mu - 1 \geq 0$, which implies $\mu \geq 1$ and $\mu \leq -1$, a contradiction. Therefore, the assumption was false, and the proposition is proved.
Q.E.D.

Theorem 3.10 *Let* **card** $(P_+) > 0$, **card** $(P_-) > 0$, *and* $\nu > 0$. *Then for the model considered in (6) the following matrix system has a unique solution*

$$\begin{bmatrix} D_a + D_b(\frac{1}{\nu}I + DAA^TD^T) & -D_bDe \\ -e^TD^T & 0 \end{bmatrix} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

for all D_a *and* D_b *defined in Theorem 3.7 and arbitrary* r_1 *and* r_2.

Proof: Since $D_a \geq 0$ and $D_b \geq 0$ with $D_a + D_b$ positive definite by Theorem 3.7, it follows from $\nu > 0$ that $D_a + \frac{1}{\nu}D_b$ is also positive definite.

Let $\bar{D} := D_a + \frac{1}{\nu}D_b$, and note that $I + A^TD^T\bar{D}^{-1}D_bDA$ is a symmetric positive definite matrix and therefore invertible. Hence, $\bar{D} + D_bDAA^TD^T$ is invertible with the inverse defined by the Sherman-Morrison-Woodbury identity as

$$(\bar{D} + D_bDAA^TD^T)^{-1} = \bar{D}^{-1} - \bar{D}^{-1}D_bDA(I + A^TD^T\bar{D}^{-1}D_bDA)^{-1}A^TD^T\bar{D}^{-1}$$

and

$$x = (\bar{D} + D_bDAA^TD^T)^{-1}(D_bDe\mu + r_1).$$

Substituting x out of the system, we are left with

$$e^TD^T(\bar{D} + D_bDAA^TD^T)^{-1}(D_bDe\mu + r_1) = r_2,$$

which simplifies to

$$e^TD^T(\bar{D} + D_bDAA^TD^T)^{-1}D_bDe\mu = r_2 - e^TD^T(\bar{D} + D_bDAA^TD^T)^{-1}r_1.$$

We now show that $e^TD^T(\bar{D} + D_bDAA^TD^T)^{-1}D_bDe$ is invertible.

Using the Sherman-Morrison-Woodbury identity for the inverse, we have the following matrix:

$$e^TD^T(\bar{D}^{-1} - \bar{D}^{-1}D_bDA(I + D^TA^T\bar{D}^{-1}D_bAD)^{-1}A^TD^T\bar{D}^{-1})D_bDe.$$

Let $\hat{D} := \bar{D}^{-1}D_b$, and rewrite this system as

$$e^TD^T(\hat{D} - \hat{D}DA(I + A^TD^T\hat{D}DA)^{-1}A^TD^T\hat{D})De.$$

Since $\hat{D}$ is a diagonal matrix with nonnegative diagonals, we can replace $\hat{D}$ with $\hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}}$ to obtain the system

$$\begin{aligned} e^T D^T (\hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} - \hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} DA(I + A^T D^T \hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} DA)^{-1} A^T D^T \hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}}) De \\ = e^T D^T \hat{D}^{\frac{1}{2}} (I - \hat{D}^{\frac{1}{2}} DA(I + D^T A^T \hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} DA)^{-1} A^T D^T \hat{D}^{\frac{1}{2}}) \hat{D}^{\frac{1}{2}} De \\ = e^T D^T \hat{D}^{\frac{1}{2}} (I + \hat{D}^{\frac{1}{2}} DAA^T D^T \hat{D}^{\frac{1}{2}})^{-1} \hat{D}^{\frac{1}{2}} De, \end{aligned}$$

where the last equality comes from the Sherman-Morrison-Woodbury identity.

The inner term, $(I + \hat{D}^{\frac{1}{2}} DAA^T D^T \hat{D}^{\frac{1}{2}})^{-1}$, is a symmetric positive definite matrix. Furthermore, since $\mathbf{card}\,(P_+) > 0$, $\mathbf{card}\,(P_-) > 0$ by assumption, it follows that $D_b \neq 0$. Hence, $\hat{D}^{\frac{1}{2}} \neq 0$ and $e^T D^T \hat{D}^{\frac{1}{2}}$ has full row rank. Therefore, (12) is symmetric positive definite and invertible. If follows that (12) is invertible and, therefore, μ and x are uniquely determined for any r_1 and r_2. **Q.E.D.**

We know that the Newton direction exists for all z and all choices of D_a and D_b. We now need to show that the sequence generated by Algorithm 3.4 has an accumulation point.

Theorem 3.11 *Suppose that* $\mathbf{card}\,(P_+) > 0$ *and* $\mathbf{card}\,(P_-) > 0$. *Then Algorithm 3.4, applied to problem (6), has an accumulation point.*

Proof: Our proof is adapted from [27]. We shall show that the level sets of $\Psi(z)$ are bounded, and hence by the descent properties of the algorithm, there must be an accumulation point of the iterates. To show this, we show instead that $\|\Phi\|$ defined by (11) is coercive. This is sufficient to prove that the level sets of $\Psi(z)$ (where $z = (x, \mu)$) are bounded. We will extensively use the fact that if $(u \to -\infty)$ or $(v \to -\infty)$ or $(u \to \infty$ and $v \to \infty)$, then $\|\phi(u, v)\| \to \infty$.

Suppose not; that is, suppose $\|\Phi\|$ is not coercive. Let $\{\left\|x^k, \mu^k\right\|\} \to \infty$ such that

$$\left\|\Phi(x^k, \mu^k)\right\| < \infty. \tag{12}$$

Without loss we can take a subsequence for which

$$\frac{(x^k, \mu^k)}{\|x^k, \mu^k\|} \to (\bar{x}, \bar{\mu}) \neq 0.$$

Furthermore, on this subsequence

$$\frac{F(x^k, \mu^k)}{\|x^k, \mu^k\|} \to \begin{bmatrix} Q\bar{x} - De\bar{u} \\ e^T D\bar{x} \end{bmatrix},$$

where Q is the positive definite matrix multiplying x in (6).

Define $\bar{w}_i = F_i(\bar{x}, \bar{\mu})$ for $i = 1, 2, \ldots, m$. If $\bar{w}_i \bar{x}_i > 0$ for some i, then $\{x_i^k\} \to \infty$ and $Q_i.x^k - D_i i\mu^k - 1 \to \infty$. In this case, $\left\|\phi(x_i^k, F_i(x^k, \mu^k))\right\| \to \infty$ and hence $\left\|\Phi(x^k, \mu^k)\right\| \to \infty$, a contradiction to (12). Thus, $\bar{w}_i \bar{x}_i \leq 0$ for each $i = 1, 2, \ldots, m$.

Now if $\bar{w}_i < 0$ for some i, then $F_i(x^k, \mu^k) \to -\infty$, resulting in $\left\|\phi(x_i^k, F_i(x^k, \mu^k))\right\| \to \infty$, a contradiction to (12). Similarly, whenever $\bar{x}_i < 0$. Thus, $\bar{w} \geq 0$, $\bar{x} \geq 0$, and $\bar{w}^T \bar{x} = 0$.

Furthermore, if $e^T D\bar{x} \neq 0$, it follows that $\left\|\Phi_{m+1}(x^k, \mu^k)\right\| \to \infty$, also a contradiction to (12). Thus, $e^T D\bar{x} = 0$.

Note that $\bar{w}^T \bar{x} = 0$ and $e^T D\bar{x} = 0$ imply that $\bar{x}^T Q\bar{x} = 0$. Since Q is positive definite, this implies that $\bar{x} = 0$. In this case, it follows from $\bar{w} \geq 0$ that $-De\bar{\mu} \geq 0$. Now, because $\|P_+\| > 0$ and $\|P_-\| > 0$, this implies that $\bar{\mu} = 0$. However, this contradicts the fact that $(\bar{x}, \bar{\mu}) \neq 0$. Thus, $\|\Phi\|$ is coercive, and the proof is complete.
Q.E.D.

Note that Theorem 3.11 remains valid for any function ϕ that satisfies the NCP-property and the simple implications given in the first paragraph of the proof above.

Corollary 3.12 *Suppose that* $\mathbf{card}\,(P_+) > 0$ *and* $\mathbf{card}\,(P_-) > 0$. *Algorithm 3.4 applied to (6) converges, and the rate of convergence is Q-quadratic.*

Proof: We have established that Φ is strongly semismooth for this problem (F is linear) and Ψ is continuously differentiable. Furthermore, by Theorem 3.11 and Theorem 3.5, there is an accumulation point of the sequence generated by Algorithm 3.4 that is a stationary point for Ψ. Since all of the elements of the B-subdifferential are invertible by Theorem 3.10, this stationary point solves the system $\Phi(x) = 0$ by Theorem 3.6. The conclusion then follows from Theorem 3.5.
Q.E.D.

4 Implementation and Computational Results

We have shown how to reformulate the support vector machine as a mixed complementarity problem using the Fischer-Burmeister function. For this reformulation, we have shown that Algorithm 3.4 can be applied, that it converges, and that the rate of convergence is Q-quadratic. In this section we discuss the implementational details and present some computational results on a randomly generated test problem with 60 million observations where each observation measures 34 features and each feature is an integer between 1 and 10 [7].

The main computation performed at each iteration of Algorithm 3.4 is to compute the Newton direction given $\Phi(x^k)$ and $H^k \in \partial_B \Phi(x^k)$. As shown in Theorem 3.10, the required direction generation can be calculated by using

$$\begin{aligned} x &= (\bar{D} + D_b DAA^T D^T)^{-1}(D_b De\mu + r_1)e^T D^T(\bar{D} + D_b DAA^T D^T)^{-1} D_b De\mu \\ &= r_2 - e^T D^T(\bar{D} + D_b DAA^T D^T)^{-1} r_1. \end{aligned}$$

We can see that there are two common components, so we define

$$\begin{aligned} y &= (\bar{D} + D_b DAA^T D^T)^{-1} D_b De \\ z &= (\bar{D} + D_b DAA^T D^T)^{-1} r_1, \end{aligned}$$

and we are left with the equivalent system of equations:

$$\begin{aligned} x &= y\mu + z \\ \mu &= \frac{r_2 - e^T D^T z}{e^T D^T y}. \end{aligned}$$

Our procedure calculates the direction by using the Sherman-Morrison-Woodbury identity to calculate y and z simultaneously with two passes through the A matrix. Having y and z, we can then easily construct (x, μ).

To calculate H^k and $\Phi(z^k)$ is straightforward and uses two passes through the A matrix. Whenever we find an element for which $\|z_i, F_i(z)\| = 0$, we simply set $D_a = \frac{1}{2}$ and $D_b = \frac{1}{2}$. While, in this case, the resulting H^k may not be an element of $\partial_B \Phi(z^k)$, we did not have any difficulty using this definition on our test problems. We note that the theory from [1] can be used to calculate an element of $\partial_B \Phi(z^k)$ in these cases using two additional passes through the A matrix.

Therefore a complete iteration of the implemented code requires four passes through the A matrix. We expect that the major amount of time is spent calculating $(I + D^T A^T \bar{D}^{-1} D_b AD)^{-1}$, since this requires mf^2 floating-point operations, where m is typically very large. However, further inspection reveals that the number of operations is a function of the number of active elements for which $(D_b)_{ii} > 0$. By Proposition 3.8 the inactive elements (those with $(D_b)_{ii} = 0$) have $z_i = 0$ and $F_i(z) \geq 0$. For the support vector machine application, the active components at the solution correspond to the support vectors, and the number of support vectors is typically much smaller than m. Therefore, we would expect that near the solution most of the components of D_b would be equal to zero. Hence, as the iterations proceed, the amount of work per iteration should decrease as a result of the removal of the inactive components. This reduction in computational effort is similar to that found in active set methods even though the algorithm does not explicitly use an active set. Figure 1 plots the percentage of elements per iteration where $(D_b)_{ii} > 0$. Toward the beginning of the computation, all of the elements are active, leading to full cost factor/solves. In later iterations, however, the potential support vectors are reduced to 80% of the original problem data, leading to an 80% reduction in the time to perform the factor. We used a zero tolerance of 10^{-10}; that is, components for which $(D_b)_{ii} < 10^{-10}$ were treated as zero in the computations.

A comparison with the interior-point method in [7] shows that the linear algebra performed is similar. However, the semismooth method can use the reduction in linear algebra performed above, whereas that interior-point method cannot, because of the interiority condition. Furthermore, the semismooth method performs only one solve per iteration, while the (predictor-corrector) interior-

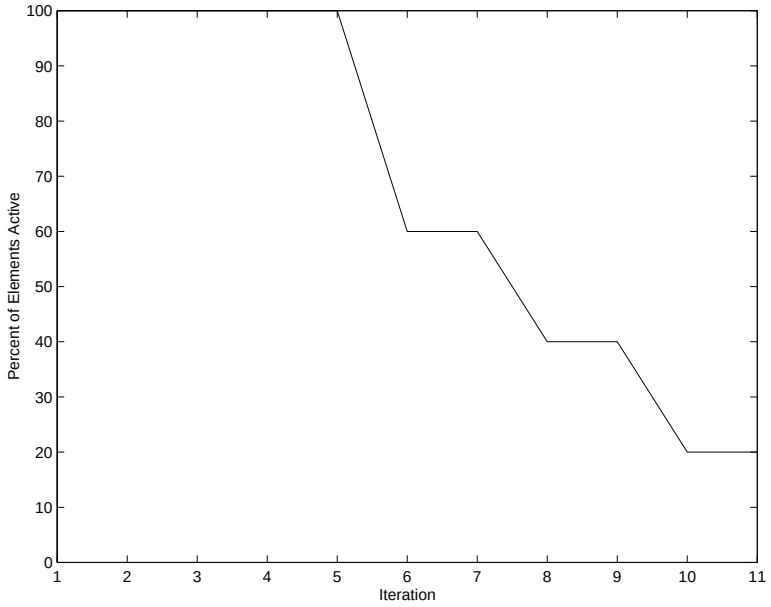


Figure 1: Percentage of observations that are active per iteration on example problem with 60 million observations and 34 features.

point method does two. We would therefore expect to obtain better performance from the semismooth method than from the the interior-point method.

The main drawback of the semismooth method is in the number of function evaluations that may be needed in order to satisfy the linesearch rule. Therefore, a nonmonotone linesearch procedure [10, 11, 5] was used within the semismooth implementation to limit the number of linesearches performed. The nonmonotone procedure allows for increases in the merit function by using a reference value in the linesearch test that decreases on average. The use of such a technique affects neither the convergence nor rate of convergence results for the algorithm. For all of the tests reported in this paper, the Newton direction was always accepted, without resorting to the linesearch procedure. Furthermore, the use of the gradient of the merit function was not encountered. As a result, the code is optimized for the case where the Newton direction is accepted. We plot the log of the residual in Figure 2 for a run using the full dataset of 60 million observations.

Our implementation of the semismooth algorithm for the support vector machine application uses a total of five vectors with n elements, one $f \times f$ matrix, and several vectors with f elements. A starting point of $(x, \mu) = 0$ was used for all tests. As indicated above, four passes through the A matrix are performed during each iteration of the algorithm. Access to the n vectors and observation matrix is provided by the low-level routines developed for the

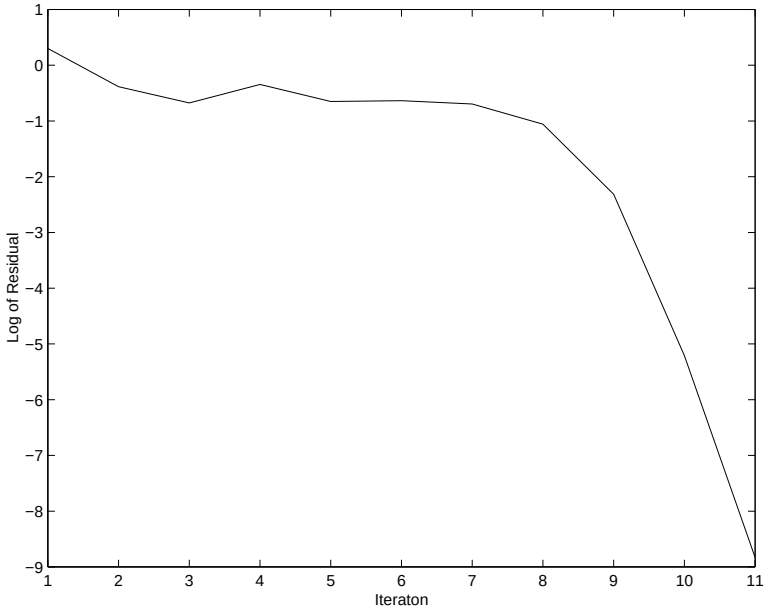


Figure 2: Log of the residual per iteration on example problem with 60 million observations and 34 features.

interior-point method in [7]. Access to the problem data (feature measurements) and vectors is provided using asynchronous I/O constructs. All of the data is stored on a large disk and sequentially accessed. While one block of data is being read from the disk, we work on the data currently available. A buffer size of 250,000 elements is used by this particular code. The buffer size corresponds to the number of rows of the A matrix kept in-core, as well as the number of elements of the vectors kept. For the interior-point method, this results in an 11% increase in time over an in-core solution for a problem with 1 million observations. For this problem, a total of 75 MB of RAM was used, which is easily accommodated by most personal computers.

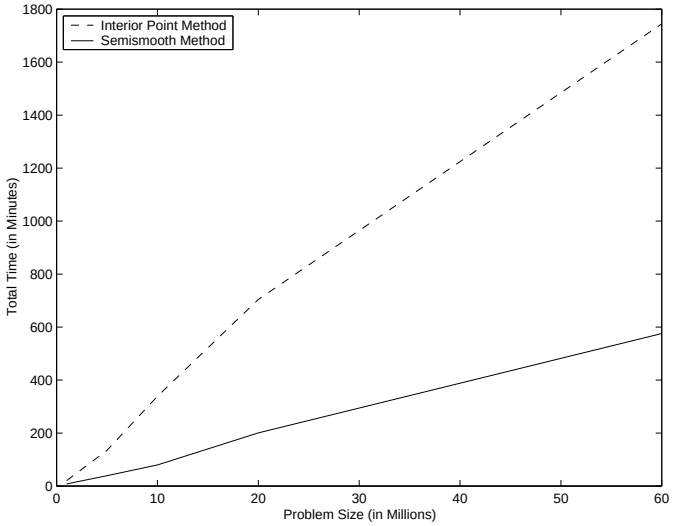
In principle, a conjugate gradient method could be used to solve the linear system of equations. However, the number of operations per solve is of the same order as our technique uses, while it is likely that many more passes through the data will be required. For this reason, we prefer the implementation described above using a direct factorization.

For the semismooth method on the 60 million point dataset, the number of iterations remained constant as we varied the number of observations selected between 1 and 60 million elements. In all cases, we performed a total of 11 function evaluations and 10 factor/solves. Furthermore, the inf-norm of the residual at the solution was between 10^{-12} and 10^{-9} for all of the solves. We used the same 60 million point data set from the interior-point methods paper [7]. In Figure 3, we plot a comparison of the number of iterations and the total time taken with the semismooth method and the interior-point method for varying problem size. We note a reduction in time of over 60% for the semismooth method. This reduction comes primarily from four sources. First, there is a reduction in the number of iterations. Second, the amount of I/O required per iteration is less for the semismooth method than for the interior-point method. Third, each iteration of the semismooth method requires one solve instead of two for the predictor-corrector interior-point code. Fourth, the work involved in factorization is reduced by the (implicit) active set nature of the semismooth method.

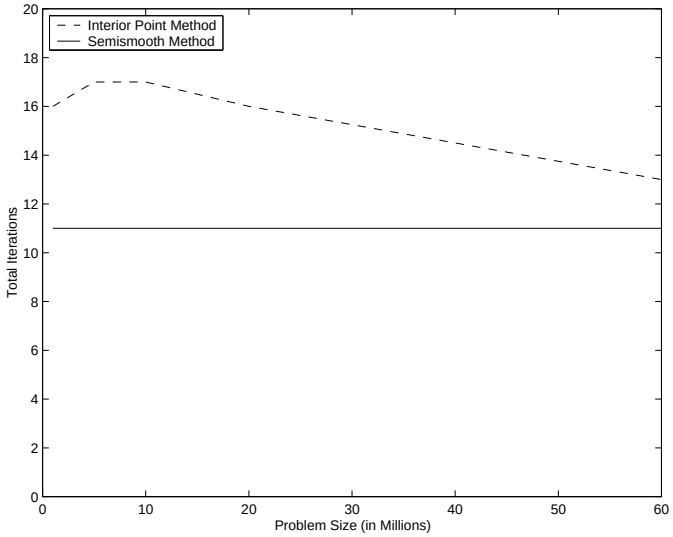
5 Conclusions

This paper presented a formulation for the support vector machine problem and proved that the semismooth algorithm converges when applied to it. These results extend the general theory to the special case of support vector machine problems.

We observe that significant reductions in the direction generation time can be obtained by using information related to the active components. Furthermore, the algorithm identifies these active components automatically. The number of active components is related to the number of support vectors in the model, which is typically much smaller than the number of variables. Our results indicate this to be the case and show substantial reductions in solution time by exploiting this fact.



(a) Total solution time



(b) Iterations

Figure 3: Comparisons between an interior-point method and the semismooth method for varying problem size.

Finally, we compared the method to an interior-point method applied to the same model and realized a significant decrease in the total solution time. We were able to solve a problem with 60 million variables on a standard workstation (using only 75 MB of RAM) in around 9.5 hours. Parallel implementations of the code are also possible, but we believe the main benefit to be that we do not require a large machine with many processors and a huge amount of RAM to obtain reasonable performance.

Other reformulations of the support vector machine that do not contain the linear constraint can also be used. In this case, we realize more dramatic improvements in performance when compared with the interior-point method on the same model. This formulation was not discussed here, as the theory is uninteresting. For completeness, we just note that removing the linear constraint gives a different model that is simply a bound constrained positive definite quadratic program. When the semismooth method is used, the amount of computation and number of iterations is about the same with or without the linear constraint. However, for the interior-point code, the number of iterations grows with problem size when there is no constraint, resulting in a quadratic (as opposed to linear) growth in time. Other techniques have been described for this case in [16].

We remark also that while the semismooth reformulation given here outperforms the interior-point method detailed in [7], the latter method is more general in that even more formulations can be solved. A key requirement for the interior-point method is that the general linear constraint matrix must have full row rank. Some formulations of the support vector machine problem become more difficult with the semismooth method because they involve a positive semidefinite quadratic term with two constraints. The proof technique presented in this paper does not apply to this case. Even if we had a positive definite quadratic term, we can demonstrate points where every element of the B-subdifferential is rank deficient, even though the constraints have full row rank.

References

[1] S. C. Billups. *Algorithms for Complementarity Problems and Generalized Equations*. PhD thesis, University of Wisconsin–Madison, Madison, Wisconsin, August 1995.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[3] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York, 1983.

[4] T. De Luca, F. Facchinei, and C. Kanzow. A semismooth equation approach to the solution of nonlinear complementarity problems. *Mathematical Programming*, 75:407–439, 1996.

[5] M. C. Ferris and S. Lucidi. Nonmonotone stabilization methods for nonlinear equations. *Journal of Optimization Theory and Applications*, 81:53–71, 1994.

[6] M. C. Ferris and O. L. Mangasarian. Breast cancer diagnosis via linear programming. *IEEE Computational Science and Engineering*, 2:70–71, 1995.

[7] M. C. Ferris and T. S. Munson. Interior point methods for massive support vector machines. Data Mining Institute Technical Report 00-05, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 2000.

[8] M. C. Ferris and J. S. Pang. Engineering and economic applications of complementarity problems. *SIAM Review*, 39:669–713, 1997.

[9] A. Fischer. A special Newton–type optimization method. *Optimization*, 24:269–284, 1992.

[10] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23:707–716, 1986.

[11] L. Grippo, F. Lampariello, and S. Lucidi. A class of nonmonotone stabilization methods in unconstrained optimization. *Numerische Mathematik*, 59:779–805, 1991.

[12] O. L. Mangasarian. *Nonlinear Programming*. McGraw–Hill, New York, 1969. SIAM Classics in Applied Mathematics 10, SIAM, Philadelphia, 1994.

[13] O. L. Mangasarian. Machine learning via polyhedral concave minimization. In H. Fischer, B. Riedmueller, and S. Schaeffler, editors, *Applied Mathematics and Parallel Computing - Festschrift for Klaus Ritter*, pages 175–188. Physica-Verlag A Springer-Verlag Company, Heidelberg, 1996.

[14] O. L. Mangasarian. Mathematical programming in machine learning. In G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Applications*, pages 283–295, New York, 1996. Plenum Publishing.

[15] O. L. Mangasarian. Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 1:183–201, 1997.

[16] O. L. Mangasarian and D. R. Musicant. Lagrangian support vector machines. Technical Report 00-06, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 2000.

[17] O. L. Mangasarian and David R. Musicant. Active set support vector machine classification. Technical Report 00-04, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 2000.

[18] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43:570–577, 1995.

[19] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23:1 & 18, 1990.

[20] R. Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization*, 15:957–972, 1977.

[21] T. S. Munson, F. Facchinei, M. C. Ferris, A. Fischer, and C. Kanzow. The semismooth algorithm for large scale complementarity problems. Mathematical Programming Technical Report 99-06, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1999.

[22] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, San Diego, California, 1970.

[23] L. Qi. Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research*, 18:227–244, 1993.

[24] L. Qi and D. Sun. A survey of some nonsmooth equations and smoothing Newton methods. In A. Eberhard, B. Glover, R. Hill, and D. Ralph, editors, *Progress in Optimization*, volume 30 of *Applied Optimization*, pages 121–146. Kluwer Academic Publishers, Dordrecht, 1999.

[25] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58:353–368, 1993.

[26] B. Schölkopf, C. Burges, and A. Smola, editors. *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[27] P. Tseng. Growth behavior of a class of merit functions for the nonlinear complementarity problem. *Journal of Optimization Theory and Applications*, 89:17–37, 1996.

[28] V. N. Vapnik. *The Nature of Statistical Learning Theory*. John Wiley & Sons, New York, 1996.